

**VOLUME 33 : ISSUE 1
FEB MAR : 2016**

**THE DEFINITIVE
QL MAGAZINE**

QUANTA

In this Issue:

News

Back 2 BASICS

Sprite Designer Pt. 3

MAGAZINE



QUANTA

INFORMATION ON THE ASSOCIATION

Membership of QUANTA, the independent QL user group, is by annual subscription. The Membership Secretary can supply full details. Copies of the association's constitution & annual accounts are available from the Secretary.

QUANTA Committee - 2015/2016

Chairman: Sarah D. Gilpin 181, Urmston Lane, Stretford, Manchester. M32 9EH
Email: chairman@quanta.org.uk Tel: 0161 865 2872

Secretary: Alison Southern 40, Distaff Road, Poynton, Cheshire. SK12 1HN
Email: secretary@quanta.org.uk Tel: 01625 850067

Treasurer & Membership Secretary: John Gilpin 181, Urmston Lane, Stretford, Manchester. M32 9EH
Email: treasurer@quanta.org.uk
Email: membership@quanta.org.uk Tel: 0161 865 2872

Magazine Editor: Lee Privett. 25 Mucking Hall Road, Barling Magna, Southend on Sea, Essex. SS3 0NH.
Email: editor@quanta.org.uk Mobile: 07985 513 234

Librarian: David Buckley 136, Denton Lane, Chadderton, Oldham. OL9 8PS
Email: librarian@quanta.org.uk Tel: 0161 622 1600

Helpline & News Editor: Dilwyn Jones 22 Erw Las, Coetmor New Road, Bethesda, Gwynedd. LL57 3NN
Email: helpline@quanta.org.uk
Email: news@quanta.org.uk Tel: 01248 601599

Web Administrator & Web Master: Keith Dunbar 44, Dalton Avenue, Stretford, Manchester. M32 9TP
Email: webadmin@quanta.org.uk Mobile: 07789 483 800

I.C.M.: Dave Park, 1406 Dana Court, Round Rock, TX USA 78664
Email: dave@quanta.org.uk Tel: (512) 529 7863

Back Issues Requests: Email: backissues@quanta.org.uk

QUANTA reserves the right to publish, reject or edit any material submitted. The opinions expressed are solely those of the contributors. Member's private small ads - free (up to 50 words). Non-members private small ads (and members after 50 words) 50p per 10 words or part thereof. Terms strictly paid up-front with order. Receipt if requested. All copyrights and trademarks are hereby acknowledged.

CONTENTS

EDITORIAL	LEE PRIVETT	4
QUANTA NEWS	DILWYN JONES	5
QUO VADIS DESIGN	ADVERTISEMENT	8
GAMING IN S*BASIC(1)	JOHN SOUTHERN	15
RWAP SOFTWARE	ADVERTISEMENT	16
JOCHEN MERZ SOFTWARE	ADVERTISEMENT	22
SMALL ADS.....	ADVERTISEMENT	32
COMMENT	GEORGE GWILT.....	39
RWAP MEMBRANES	ADVERTISEMENT	40
NEMQLUG NOTICES.....	SARAH GILPIN.....	41
SPRITE DESIGNER PT. 3....	LEE PRIVETT	42
COPY DATE	(FOR NEXT ISSUE)	51
QL FORUM	ADVERTISEMENT.....	52



Visit our website at : www.QUANTA.org.uk

'Like' us on Facebook at :

www.Facebook.com/QUANTA.org

H

ello fellow QLER's, welcome to QUANTA 2016 proper and to this years makeover. You will see we are attempting to revitalise some basic QL programming with our readership and spur you all on to tinker, experiment and play with the examples we put forward.

By all means please submit anything you wish to, we are ALWAYS needing articles for the magazine.

For those of you who partake of frequent cinema going, may have recently seen PIXELS. This Adam Sandler film is a take on the Aliens invading Earth genre where the aliens took our early computer based games as a declaration of war and also the method by which to fight battles.

It's a fun light-hearted easy going film but it's the graphics that interested me. They have used very high quality variation of pixelated images in 3D reminiscent of original arcade games way back when the QL was in its infancy. Check out the trailer on Youtube for an example. Pacman is of particular interest, to say anymore would invoke SPOILERS so I will leave it there.

Enjoy the magazine, comments as usual through the normal channels, page 2 for details.

The Editor



If you have QL-related news items that you'd like us to include on this page, please get in touch with News Editor - Dilwyn Jones at news@quanta.org.uk

GOLD & SUPER GOLD CARD REPLACEMENT BATTERIES

Between mid-December and mid-January, some battery replacements for these popular QL expansions were made available through SellMyRetro.com by Paul Veltjens in Germany, who builds and supplies the QL-SD hardware.

This is a plug in replacement for the red SAFT 40 LF 220 3V Lithium Battery (40LF220) that used to backup the Goldcard clock. It has a holder for a CR2032 lithium battery (which is not included with the package). The battery holder costs just £5.00 plus cost of shipping the device. It is not yet known when further batches of this handy little device will be made available.



The Gold/Super Gold Card battery holders (picture from SellMyRetro.com)

JPEG, GIF AND NOW PNG

David Westbury has issued the latest update to his graphics conversion software. The latest version of the functions for converting JPEG and GIF graphics to QL PIC files can now also handle the PNG graphics file formats.

Also included are utilities to extract file information from these graphics file formats, along with a small utility to create wallpaper format files (BGIMAGE in SMSQ terminology – essentially a PIC file without the 10 byte preamble).

Being written as BASIC extension functions, these graphics conversion utilities are easy to add to your programs, including compiled BASIC programs.

The author says that having developed these extensions as stand alone utilities for testing and development purposes, he now intends to move on to update the original Photon software. Download FJPEG, FGIF and FPNG free of charge from

<http://www.dilwyn.me.uk/graphics/index.html>

Scroll down the page to the entry for Photon, and this software may be found just under that.

ASSEMBLY LANGUAGE eMAGAZINE 3

Norman Dunbar writes:

After many months of the odd hour grabbed here and there, between work, driving, home life and so forth - did I mention Christmas, New Year and a holiday to boot - the latest somewhat exciting issue of the Assembly Language eMagazine is now available for download.

Point your browsers at

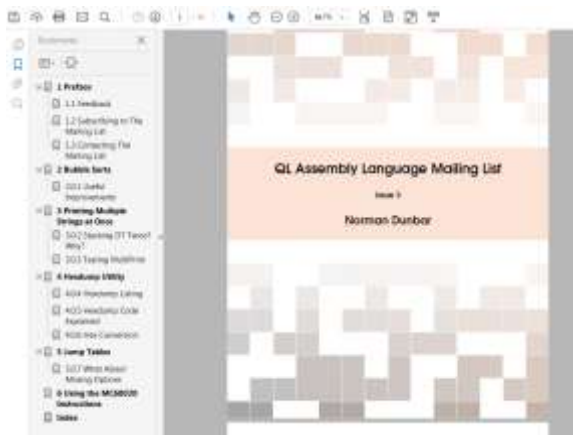
http://qdosmsq.dunbar-it.co.uk/downloads/AssemblyLanguage/Issue_003/Assembly_Language_003.pdf

(or wget

http://qdosmsq.dunbar-it.co.uk//downloads/AssemblyLanguage/Issue_003/Assembly_Language_003.pdf)

and all will be revealed.

We have 29 pages of articles on sorting data, printing multiple strings, a hex dump utility, all you never needed to know about using jump tables in your code, and some information about upcoming articles on the (new) 68020 instructions available in QPC, but sadly, not in any of the other emulators - yet. Happy reading and hopefully, the next issue will be out much quicker than this one!



Screen Shot from the PDF

QUO VADIS
DESIGN

Independent Information
Technology Services

www.ql-qvd.com

QL/QDOS/SMSQ/E Software

QUO VADIS DESIGN Independent Information Technology Services

QL/QDOS/SMSQ/E Software

Home Products Support Company Contact

Welcome


Quo Vadis Design sells software for the Socratic Quantum Leap computer (QL) and variants including a true OS called SMSQ/E.

The QL is a computer in its 25th year Anniversary.

News

- QVD QL News Blog - keep up to date
[News Blog](#)
22/04/2009
- Quo Vadis Design Website Launched
21/02/2009

FEATURED PRODUCT



BUY NOW!

Copyright © 2009 Quo Vadis Design. All Rights Reserved

Home | Products | Support | Company | Contact

Bruce@ql-qvd.com

Quo Vadis Design
38 Derham Gardens
Upminster
RM14 3HA
UK

Tel: +44 (0)20 71930539
Fax: +44 (0)870 0568755

**QPC2 IS NOW FREE AND
AVAILABLE FROM MARCEL
KILGUS WEBSITE**

<http://www.kilgus.net/qpc/index.html>

**Congratulations to Marcel
on becoming a Father**

**Check the QL News Blog on
our website for updates.
www.ql-qvd.com/blog**

SMSQMULATOR V2.16 RELEASED

Wolfgang Lenerz reports that he has released version 2.16 of the Java-based QL emulator called SMSQmulator. Recent updates to this system include:

- V2.14 a small bug fix which could arise when using beep with openJDK.
- V2.15 various small bug fixes.
- V2.16 Small optimizations for speed, screen update interval is selectable.

Download SMSQmulator free of charge from Wolfgang's website at

<http://www.wlenerz.com/SMSQmulator/>

Versions for Java 6, 7 and 8 are available.

IP MANUAL

Martin Head has put together a document to aid in programming in assembler, that he has cobbled together as he has been going along learning to program TCP/IP on QDOS and SMSQ, from documents by Richard Zidlicky and Jonathan Hudson, and from 'C' code and information from the internet.

It lists and describes the IP system calls, and data structures used. It also includes a list of 'C' error codes and descriptions to help you when debugging machine code programs.

The document is unfinished, and still very much a work in progress, so it's bit patchy. However he has made it available in case anyone is interested in programming the IP drivers in assembler, or to help if anyone wants to understand his Client/Server program.

The document is intended to be printed as a booklet on A5 paper and at the time of writing consisted of about 64 pages.

QL Forum members can access the document from

<http://www.qlforum.co.uk/viewtopic.php?f=3&t=1403&start=30>

SFERA

As part of his efforts to further the use of Forth on QL systems, Marcos Cruz has released software called SFERA, which stands for SuperForth Extensions, Resources and Add-ons. It's a library for SuperForth, a Forth-83 system for Sinclair QL, written by Gerry Jackson and published by Digital Precision in 1985.

Marcos writes:

I started writing SFERA in order to make the development of some new projects easier. Some of the main goals of SFERA are the following:

- Many common words and tools found in modern standard Forth systems.
- Improved support for source text files, e.g. nesting and dependencies.
- Words for accessing some features of the SMSQ/E operating system.
- Improved blocks handling, e.g. dependencies.

At the time of writing (15th January) SFERA was described as “barely usable”, although it was a work in progress. Gerry Jackson has put on GitHub the sources of his almost finished QL SuperForth cross-compiler, which he started based on the original SuperForth. Most of the sources are common, so it's most useful to understand SuperForth.

<https://github.com/gerryjackson/QL-SuperForth>

Probably also the sources of the "real" SuperForth, published by Digital Precision in 1985, will be rescued from the original 5.25 floppy disks and published.

Gerry Jackson has also put on GitHub the sources of his game Reversi, written in SuperForth. It's version 1.8, newer than the one originally published in 1985.

<https://github.com/gerryjackson/QL-Reversi>

There are chess and go games written in Forth but as far as Marcos knows there's no other Reversi in Forth. Perhaps some day someone will convert it to standard Forth or other Forth system. Probably the sources of version 1.2, which included many comments, will be restored from the scanned manual.

Follow QL and Spectrum Forth developments on the Forth mail list reported in the Dec 2015/Jan 2016 issue of QUANTA magazine (page 13):

<http://programandala.net/en.forth-sinclair.html/> (English)

<http://programandala.net/es.forth-es.html/> (Spanish)

PROSPERO QL FORTRAN MANUAL

Derek Stewart has taken the scanned Fortran manual available from my website and drastically reduced the file size down from about 67MB to about 3.5MB. Both the original and reduced versions are available to download free from my Languages page at:

<http://www.dilwyn.me.uk/language/index.html>

URS KÖNIG WEBSITE CHANGE

Urs König's website has moved from :

<http://www.qlvsjaguar.homepage.bluewin.ch>

to a new address :

<http://www.sinclairql.net/>.

30 YEARS OF URS

Urs König writes:

30 years ago today (20th January) I bought my 1st personal Sinclair QL Professional Computer. This changed my life in many ways. To celebrate this I've spent an hour today to re-publish the QL chronology webpage which has been taken down by the ISP Swisscom last autumn due to end of service of their xy.homepage.bluewin.ch hosting package.

<http://sinclairql.net/chronology.html#MyFirstQL>

DATES TOOLKIT SOURCES

Per Witte writes:

Someone on QL-forum requested the sources for the Dates toolkit (as found on Toolkits page on Dilwyn's site). I don't have compact copies of the sources (they're spread around and would require actual work to make into a compilation), only the HTML-ised, navigable versions off my old web site.

So with the license, meta info and the binaries, it all boils down to about 40Kb. Dates001.zip is available from:

www.dilwyn.me.uk/tk/index.html

Download, Unzip somewhere and click on Index.html

TOOLKIT 2 V2.23 AND V2.26

David Westbury has managed to extract copies of Toolkit 2 from two different Trump Card ROM versions. Both versions seem to be level 2 filing system aware (although David has only been able to test them with Miracle hard disk drive) and come with source assembler. Both have been given suitable headers to allow them to be loaded to RESPR area. Note that the header makes the toolkit size just over 16K in length.

Download from <http://www.dilwyn.me.uk/pe/index.html>

SER_USB V2 DRIVERS

Thanks to Urs König, I have been able to make available version 2 of the drivers for the short-lived SER-USB add-on produced by Memory Lane Computing. Available to download from:

<http://www.dilwyn.me.uk/misc/index.html>

68K PROCESSORS AVAILABLE AGAIN

(This is a post from QL Forum by Thorsten Herbert)

Rochester brings Freescale 68K, Intel 80C186/88 MCUs back to life. Rochester Electronics has worked out agreements with Freescale and Intel to bring various members of the 68000 and 80C186/88 embedded microcontrollers back into production for continued availability.

The company specializes in bringing back into production mature and end-of-life semiconductor devices and will next year (2015) add the Freescale and Intel processors to its current roster of such end of life designs.

According to Paul Gerrish, President at Rochester Electronics, the company has acquired all of both company's remaining finished devices and wafer/die as well as intellectual property in order to manufacture the exact same device and provide a reliable continuing source of the devices for systems that continue to use these semiconductors.

He said the Freescale 68020 processor is available now, with the full military version of the 68020 in production by the first quarter of 2015. The 68020 processor was sampled in 2014 and will ramp up production in the first quarter of 2015.

Gerrish said plans are in the works for the rest of the 8-bit NMOS family of products featuring the 6821, 6840, and 6850 in addition to the 6809. In addition to these products, the 68HC000 family and the 68882 are also in development. Also, other Freescale microcontrollers, such as the 68HC05 and 68HC11, are scheduled for development in 2015 he said.

"Intel products such as the 80C186EA, EB, EC, XL, and the 80C188EA, EB, EC and XL are all into fabrication now," he said. "The EB is currently ready for qualification. Also in the development pipeline for 2015 are the Intel 8X196KB, KC, and KD microcontrollers."

MORE OLDER GAMES AVAILABLE

Thanks to Detlef Obermann, I have been able to make available a number of older games sourced from German QL PD disks. The titles include Super Star Trek, Starport, Space Pods, Q-Slot, QL-Zeitfalle (Time Event), Rescue Of Horan, Fred (Poker, Black Jack and High Low), Gambler (Backgammon, Checkers, Nine Men's Morris) and German version of Grey Wolf (English language

version available from RWP Software) and Final Conflict. Get all these free of charge from the Games page on my website at <http://www.dilwyn.me.uk/games/index.html>

GRAF3D

Also rescued is a program called Graf3D, which renders your Abacus _exp files into 3D graphs. By Hans-Gerd Peerenboom 1990, it comes with a German _doc file. It's a mere 30KB download from <http://www.dilwyn.me.uk/graphics/index.html>

GAMING IN S*BASIC (1) JOHN SOUTHERN

T

o celebrate our QUANTA subgroup when it was 21 years old we decided to try and have a project. We settled on trying to write a couple of games for a QL that would work on an unexpanded system, in SuperBASIC and if possible then try to change these to run under the pointer environment.

Part of the idea was we would just have fun trying to get something working on screen.

The first was a version of 2048. This is a very simple sliding puzzle on a 4x4 grid where your only options are the arrow keys and everything slides in that direction. We start with just two squares filled with a number (either 2 or rarely 4). Each slide causes another number to appear in one of the free grid locations. If two of the squares are the same number, when they slide and hit one

RWAP Software

We have been working hard to safeguard the future of the Sinclair QL through creating an online QL Wiki backed by a repository of QL titles.

The QL Wiki enables people to see what software and hardware was produced for the Sinclair QL. The repository enables people to purchase working copies when their microdrive cartridges fail, or they move onto disks or emulators. Where we can, we obtain permission from the copyright holders to re-release items on the market.

This has enabled us to bring some of the titles back to the market as well as re-kindle interest in the Sinclair QL and its emulators.

Old favourites brought back to life include:

E.V.A., Vroom
Black Knight, Double Block
Jungle Eddi, The Lost Pharaoh
Pangl, the Lonely Joker
Microdeal's Flight Simulator, QL Hopper, The King,
Night Nurse and Cuthbert in Space
Mortville Manor, Darkside of the Moon [£10]

We still retain our existing catalogue, including:

Open Golf, Return to Eden,
Stone Raider II, The Prawns,
Hoverzone, Lost Kingdom of Zkul,
Deathstriks, Horrorday,
Flightdeck and QWord

All games are available on disk or for use with Q-emuLator on the PC **from ONLY £5 each**

Microdrive versions also available - from £10 each.
Memory / system limits may apply - please check before ordering.

Visit the QL Wiki for more details on software, books and hardware for the Sinclair QL:

www.rwapadventures.com/ql_wiki

RWAP Software
3 Dale View Court, Fulford, Stoke-on-Trent
Staffordshire ST11 9BA
Tel: 01782 398143
Email: rich@rwapservices.co.uk

Website:
www.rwapsoftware.co.uk or www.sellmyretro.com



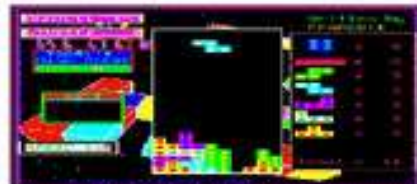
Vroom by Pyramide



E.V.A by Westway



Mortville Manor by Pyramide



Double Block by Francois Lanreut



Night Nurse by Microdeal

another, they combine to remove one square from the grid and the remainder doubles in value.

The game is usually found online to play at:

<http://gabrielecirulli.github.io/2048/>

For a SuperBASIC program we can split this program down into a few procedures.

For our main game play we will have a matrix 4x7. Knowing that we count from zero we have the line

```
300 DIM a(3,6)
```

Why 7 when we only have a 4x4 grid? Let us build up the routine to find out. To handle the movement we actually only consider sliding everything to the left. If we have pressed the up key we could either work out how to slide everything up or we could rotate our whole grid once counter-clockwise, do the left slide and then rotate back. Actually as we will build a routine to rotate counter-clockwise, we could just rotate another three times counter-clockwise rather than once clockwise.

This way a right key press means we rotate twice, slide to the left and then rotate twice again and finally a down key means we rotate three times, slide and then rotate a final time. That means we only need to write one set of procedures to handle the movement, one to rotate the matrix and call this the correct number of times before we slide and after depending on the key press.

Breaking down the movement a little further we need to slide the squares on each row to the left if there are any blank squares. We then check to see if two are the same and if so we combine and

finally slide again just in case this has produced a hole.
The worst case is when we have Space 2 2 2.

The first slide gives us 2 2 2 space

The add now give us space 4 2 space

A final slide gives 4 2 space space

We could have a counter that for each space that we have slipped over we add another space at the end, but a simpler way is to have a matrix that is seven cells long where the final three are always spaces.

In our slip routine we will work through each row of our matrix in turn so we use a FOR NEXT loop

```
720 DEFine PROCedure slip
730 FOR b=0 TO 3
...
870 NEXT b
880 END DEFine
```

Remembering that we always count from zero. Inside this loop we will next work through each cell column in turn to check if it is empty so we have another FOR NEXT loop. We only need to do this three times as if the last cell is a space it does not matter.

```
720 DEFine PROCedure slip
730 FOR b=0 TO 3 : REMark Work through the ROWS
740 FOR c=0 TO 2 : REMark Work through the
```

COLUMNS

...

860 NEXT c

870 NEXT b

880 END DEFine

We will test to see if the cell is empty

```
760 IF a(b,c)=0 THEN LET s=1
```

If it is empty we will move everything in the next columns down one.

```
770 IF s=1 THEN LET a(b,c)=a(b,c+1)
```

```
780 IF s=1 THEN LET a(b,c+1)=a(b,c+2)
```

```
790 IF s=1 THEN LET a(b,c+2)=a(b,c+3)
```

```
800 IF s=1 THEN LET a(b,c+3)=a(b,c+4)
```

That just leaves us to note if we did have a number in any of the cells we will want to add a new number randomly into the grid later. So we test if we had an empty cell ($S=1$) and something greater than 0 in one of the other cells. If this is true we just set a counter n to 1 for later.

```
810 IF s=1 AND a(b,c)>0 THEN LET n=1
```

```
820 IF s=1 AND a(b,c+1)>0 THEN LET n=1
```

```
830 IF s=1 AND a(b,c+2)>0 THEN LET n=1
```

There is the case where we had space, space, space, number.

Here we actually need to test the first space more than once so I

put in another loop to do this test three times.

```
750 FOR e=0 TO 2 : REMark Work through in case we
have up to three spaces before a number
...
850 NEXT e
```

Our whole routine is now

```
720 DEFine PROCedure slip
730 FOR b=0 TO 3 : REMark Work through the ROWS
740 FOR c=0 TO 2 : REMark Work through the
COLUMNS
750 FOR e=0 TO 2 : REMark Work through in case we
have up to three spaces before a number
760 IF a(b,c)=0 THEN LET s=1
770 IF s=1 THEN LET a(b,c)=a(b,c+1)
780 IF s=1 THEN LET a(b,c+1)=a(b,c+2)
790 IF s=1 THEN LET a(b,c+2)=a(b,c+3)
800 IF s=1 THEN LET a(b,c+3)=a(b,c+4)
810 IF s=1 AND a(b,c)>0 THEN LET n=1
820 IF s=1 AND a(b,c+1)>0 THEN LET n=1
830 IF s=1 AND a(b,c+2)>0 THEN LET n=1
840 LET s=0
850 NEXT e
860 NEXT c
870 NEXT b
```

```
880 END DEFine
```

Note in line 800 we have a C+4 and with the loop at line 740, C could be 2 so the maximum value of cells across would be 2+4=6 but as we start counting from 0 that is the seventh cell and the reason why we have a 4x7 matrix.

Let us move straight on to the add procedure. This is where we have slipped the row to the left and now two cells are the same. These should combine. Again we will work through each row in turn so we have a FOR NEXT loop

```
890 DEFine PROCedure add
900 FOR b=0 TO 3 : REMark Work through the ROWS
...
990 NEXT b
1000 END DEFine
```

We want to test the cells on our row and we need to start with the first three cells and compare to the next cell

```
910 FOR c=0 TO 2 : REMark Test each of the first three cells to
the next
...
980 NEXT c
```

We test to see if they are the same and if so we set a counter running ($g=1$). If the counter is true we make our cell become a space and double the next cell (Lines 930 and 940). We also advance our counter an extra jump at line 960 so in the case where we have three cells the same we do not add all of them. Should we have all four cells the same this leads us to have

JOCHEN MERZ SOFTWARE

Kaiser-Wilhelm-Str. 302
47169 Duisburg, Germany

Fax +49 203 502012
EMail: SMSQ@J-M-S.com

SMSQ.J-M-S.COM

Thanks to Marcel, QPC2 is now freely available.
You can download it from Marcells homepage

www.Kilgus.net

If you wish to print from QPC2, then you need

QPCPrint

... which is available from J-M-S.

Only 39.90 if you choose EMail delivery.

For additional 4 EUR, delivery will be on CD.

QPCPrint will allow you to print to (more or less) every printer which is installed under Windows (dot matrix, ink, laser, PDF "printer", FAX "printer" etc.)

You can place your order via letter or use the SSL order form on SMSQ.J-M-S.COM - click on "Online orders".

If you order by Mail or through the SSL contact form: We now accept VISA, MasterCard, Diners Club, JCB, Discover, UnionPay, BCard, DinaCard and American Express!

The order form has not been updated for the new card types at the time I write this ad, but it will be updated soon.

I am still working on changing the forms and implementing SEPA on all the order forms.

space, doubled number, space, doubled number.

Line 950 adds our new number to the score total

```
920 IF a(b,c)=a(b,c+1) THEN LET g=1 : REMark Test  
if the cells are the same
```

```
930 IF g=1 THEN LET a(b,c)=0
```

```
940 IF g=1 THEN LET a(b,c+1)=2*a(b,c+1) : REMark  
Double the cell contents
```

```
950 IF g=1 THEN LET q=q+a(b,c+1) : REMark We let  
a score Q increase
```

```
960 IF g=1 THEN LET c=c+1 : REMark Jump a cell if  
we have just added two
```

```
970 LET g=0
```

So our finished routine is

```
890 DEFine PROCedure add
```

```
900 FOR b=0 TO 3
```

```
910 FOR c=0 TO 2
```

```
920 IF a(b,c)=a(b,c+1) THEN LET g=1
```

```
930 IF g=1 THEN LET a(b,c)=0
```

```
940 IF g=1 THEN LET a(b,c+1)=2*a(b,c+1)
```

```
950 IF g=1 THEN LET q=q+a(b,c+1)
```

```
960 IF g=1 THEN LET c=c+1
```

```
970 LET g=0
```

```
980 NEXT c
```

```
990 NEXT b
```

```
1000 END DEFine
```

Our next routine is the one where we rotate our matrix. We use a second matrix to temporary hold our data and so we loop through each row and each column and write this to our new matrix.

```
1010 DEFine PROCedure transform
1020 FOR b=0 TO 3
1030 FOR c=0 TO 3
1040 LET d(b,c)=a(b,c) : REMark Copy the matrix
into a dummy holding one
1050 NEXT c
1060 NEXT b
...
1120 END DEFine
```

We then work through the new matrix and write back to our original matrix but rotated 90 degrees

```
1070 FOR b=0 TO 3
1080 FOR c=0 TO 3
1090 LET a(b,c)=d(c,3-b) : REMark This time we
have swapped b and c and count backwards
1100 NEXT c
1110 NEXT b
```

Our next procedure is the largest and is used to insert a new number into an empty cell. We need to start off by counting the

number of empty cells and holding that value in variable z.

```
1130 DEFine PROCedure inset
1140 FOR b=0 TO 3 : REMark Work through the ROWS
1150 FOR c=0 TO 3 : REMark Work through the
COLUMNS
1160 IF a(b,c)=0 THEN LET z=z+1 : REMark Increase
the number of empty cells
1170 NEXT c
1180 NEXT b
...
1330 END DEFine
```

We now use this total of empty cells to pick one at random

```
1190 LET p=INT(RND(1,z))
```

And now we find that empty cell again and put a number 2 in it.

```
1200 FOR b=0 TO 3
1210 FOR c=0 TO 3
1220 IF a(b,c)=0 THEN LET l=l+1
1230 IF l=p AND a(b,c)=0 THEN LET f=1
1240 IF f=1 THEN LET a(b,c)=2
1260 LET f=0
1270 NEXT c
1280 NEXT b
```

Rarely we need to change this to a 4 rather than a 2. We will choose to insert a value of 4 for 20 per cent of the time

```
1250 IF f=1 AND INT(RND(1,10))>8 THEN LET a(b,c)
=4
```

In the case where there had only been one single space which we have filled we run another procedure to do a final check to see if the game has ended and also reset our count of spaces, both z and l (which in hindsight we could have used the same variable as it is just a count) and w which we use in our end test to indicate the end of the game.

```
1290 IF z=1 THEN endcheck
1300 LET l=0 : REMark The number of empty cells
1310 LET z=0
1320 LET w=0
1330 END DEFine
```

For the endcheck procedure, we only use this test when we have filled all the cells, so we are just testing to see if we still have two cells next to each other holding the same value. If they do we could slide and combine, but if every adjacent cell is different then the game is over.

```
630 DEFine PROCedure endcheck
640 FOR b=0 TO 3 : REMark Work through the ROWS
650 FOR c=0 TO 2 : REMark Check the first three
cells with the next cell
660 IF a(b,c)=a(b,c+1) THEN LET w=1 : REMark Work
across the cells
670 IF a(c,b)=a(c+1,b) THEN LET w=1 : REMark Work
```

down the cells

```
680 NEXT c
```

```
690 NEXT b
```

```
700 IF w=0 THEN PRINT "YOU LOSE":STOP
```

```
710 END DEFine
```

Line 660 tests within the row while line 670 tests columns. If they are the same $w=1$ and so we do not lose and come out of the procedure. If we have not found two adjacent cells the same we end the routine. In hindsight we should have had

```
705 LET w=0
```

rather than line 1320 to keep the variable within its routine.

We have moved everything around so we need to draw all this on the screen. We start by clearing the screen and then stepping through the matrix to print the value on the screen. Rather than have the screen filled with 0s we replace these with dashes. We multiply the column position by 5 so that once we are beyond single digit values in our cell it does not overwrite other cells and spaces then out nicely. Should anyone get to a cell with a value of 16384 we will need to rewrite with a column multiplier greater than 5

```
1340 DEFine PROCedure drawscreen
```

```
1350 CLS
```

```
1360 FOR b=0 TO 3
```

```
1370 FOR c=0 TO 3
```

```
1380 AT b,c*5:PRINT a(b,c)
```

```
1390 IF a(b,c)=0 THEN AT b,c*5:PRINT "-"
```

```
1400 NEXT c
1410 NEXT b
1420 AT 5,0:PRINT "score=";q
1430 END DEFine
```

That leaves us needing three more sections. Our keyboard input, our initialisation of variables and our main program loop. Our next procedure that we will look at is for keyboard input. We need to detect the four cursor keys plus the Q key so we can Quit the game if we need to.

```
530 DEFine PROCedure keys
540 REPEAT wait
550 LET y=CODE(INKEY$)
560 IF y=113 THEN EXIT wait : REMark Q
570 IF y=192 THEN EXIT wait : REMark LEFT
580 IF y=200 THEN EXIT wait : REMark RIGHT
590 IF y=208 THEN EXIT wait : REMark UP
600 IF y=216 THEN EXIT wait : REMark DOWN
610 END REPEAT wait
620 END DEFine
```

When the procedure keys is called it enters a little loop that only exits if one of our chosen keys is selected. We have used the y variable in this case so somewhere in our initiation procedure to set variables we will give this an initial value.

Now we have y to be equal to the code of our keypress (The codes of each key can be found in the QL manual Concepts section

which is available online at <http://www.dilwyn.me.uk/docs/ebooks/olqlug/index.htm> where 208 should be labelled Cursor UP). Part of the main program loop deals with calling each routine.

```
20 REPEAT main
30 drawscreen
40 keys
50 IF y=192 THEN LET r=0
60 IF y=192 THEN LET m=0
70 IF y=200 THEN LET r=2
80 IF y=200 THEN LET m=2
90 IF y=208 THEN LET r=1
100 IF y=208 THEN LET m=3
110 IF y=216 THEN LET r=3
120 IF y=216 THEN LET m=1
130 IF y=113 THEN STOP
140 FOR j=0 TO 3
150 IF r>0 THEN transform
160 LET r=r-1
170 NEXT j
180 slip
190 add
200 slip
210 IF n=1 THEN inset
220 LET n=0
230 FOR k=0 TO 3
240 IF m>0 THEN transform
```

```
250 LET m=m-1
260 NEXT k
270 END REPEAT main
```

Lines 50 to 120 set two variables depending on which arrow key is pressed. The variable r is the number of transforms we need to do to the matrix before we start to slip and m is the number after to rotate the matrix back to where it was. We could have removed all the lines that set m and added a single line LET m=4-r
Line 10 just calls a procedure to initialise all the variables
10 initial

That leaves our final procedure to ensure all variables have an initial value.

```
280 DEFine PROCEDURE initial
290 RANDOMISE
300 DIM a(3,6)
310 DIM d(3,6)
320 LET f=0
330 LET g=0
340 LET l=0
350 LET m=0
360 LET n=0
370 LET p=0
380 LET q=0
390 LET r=0
400 LET s=0
```

```

410 LET w=0
420 LET y=0
430 LET z=0
440 FOR b=0 TO 3
450 FOR c=0 TO 6
460 LET a(b,c)=0 : REMark initialise the matrix
to all zeros
470 LET d(b,c)=0 : REMark initialise the
temporary matrix to all zeros
480 NEXT c
490 NEXT b
500 inset
510 inset
520 END DEFine

```

Lines 440 to 490 just fill the matrices with 0 as a value. The two last lines 500 and 510 run the procedure to insert the two opening numbers into the grid.

I would like to be able to say that it took under thirty minutes to program but it took much longer as I had to look up some commands in the manual. Notably RND and the AT command for printing text at a specific location.

Some will have noticed some wasted parts such as

```
310 DIM d(3,6)
```

We only ever use a 4x4 grid for d to hold the temporary grid data while rotating the matrix. However, by keeping it the same size as

SMALL ADS

FOR-SALE

Three spaces around a page or more in size (either blank or bordered) for keen amateur articles (MUST BE SEEN), three available are Comments, Programming, Reviews. Offers in excess of 100 words to the Editor

!!!Worried???

Don't worry about your spelling or grammar not being up to scratch, that is something we can correct or leave as necessary, the Editor's decision is final. Why not write a review, A review of a piece of hardware you have recently seen, obtained or bought that is related or used with the Sinclair QL. A review of a piece of Software you have bought or downloaded legally from a website that doesn't stick a virus on you system. Reviews of any other media such as books, magazines or even leaflets that other people may appreciate. Contact Mr Privett for details on how you can overcome your worry.

WANTED

With FTC (Fleet Tactical Command) having been found, I'm now trying to track down FTC II by Diren which is currently missing in action. I'm also after Top Team by Arunsoft which was a football management game. If you can help please contact Peter at:

peetvanpeebles@yahoo.co.uk

LOST

The ability to do something positive, then submit something the editor, you know it makes sense.

£££ SPARE-TIME?

Earn yourself lots of gratitude with a certificate of thanks by contributing something to the magazine. We can't pay you £££s or ££s or even £ but what you will be doing will be rewarding in itself and you get your name in typeset style lights (without the lights bit). **Interested?**, then contact the editor of the magazine via the contact page inside the front cover. **Not interested?**, then contact the editor of the magazine via the contact page inside the front cover. We will be glad to hear from you.

SHEDS SHEDS SHEDS

Shed loads of space for you to write or submit that SuperBASIC listing for the magazine. **Short** Listings - any number of lines really, from just a few to a couple of pages, single or multiple procedures and functions. As long as you have written to either solve or explore a programming problem. Or even just for a bit of experimental fun. **Long** listings? No program is too big, no language too strange, whether its just concepts, SuperBASIC, block diagrams, Boolean logic, assembler, mnemonics (that's one for you George), C# C++ C - - or just plain C, Pascal, Fortran or even S*BASIC, anything about everything would be appreciated. What if you don't know if your program listing is too long to be short or too short to be long? **NO WORRIES**, we accept medium listings too throughout the year so get writing **NOW!**

Contact us in the usual way, so contact us **NOW!**

If you would like to place a small ad then go to page 2 for full details of how to go about it, options are for QUANTA members as well as non-QUANTA members.

variable a we can use the same FOR NEXT loops at 440 and 450 to initially fill them with zero values.

We separated out each variable to a separate line and could have easily combined lines 320 to 430 into just one multi-statement line. The thinking behind this was portability. By altering the drawscreen procedure and lines 1190 and 1250 which have the RND commands we were able to quickly port to other old machines such as the Mattel Aquarius. Most early computers do not seem to have procedures built into their BASIC languages but we can replace these by calling them with GOSUB and having a RETURN at the end of each.

By keeping to simple procedures we also hoped to reuse them in other games during the coming months.

For the next step I handed over the code to other members of our subgroup who took the simple game and turned it into something worth playing by changing it to be a Pointer Environment game.

```
10 initial
20 REPEAT main
30 drawscreen
40 keys
50 IF y=192 THEN LET r=0
60 IF y=192 THEN LET m=0
70 IF y=200 THEN LET r=2
80 IF y=200 THEN LET m=2
90 IF y=208 THEN LET r=1
100 IF y=208 THEN LET m=3
```

```
110 IF y=216 THEN LET r=3
120 IF y=216 THEN LET m=1
130 IF y=113 THEN STOP
140 FOR j=0 TO 3
150 IF r>0 THEN transform
160 LET r=r-1
170 NEXT j
180 slip
190 add
200 slip
210 IF n=1 THEN inset
220 LET n=0
230 FOR k=0 TO 3
240 IF m>0 THEN transform
250 LET m=m-1
260 NEXT k
270 END REPEAT main
280 DEFINE PROCEDURE initial
290 RANDOMISE
300 DIM a(3,6)
310 DIM d(3,6)
320 LET f=0
330 LET g=0
340 LET l=0
350 LET m=0
360 LET n=0
```

```
370 LET p=0
380 LET q=0
390 LET r=0
400 LET s=0
410 LET w=0
420 LET y=0
430 LET z=0
440 FOR b=0 TO 3
450 FOR c=0 TO 6
460 LET a(b,c)=0
470 LET d(b,c)=0
480 NEXT c
490 NEXT b
500 inset
510 inset
520 END DEFine
530 DEFine PROCedure keys
540 REPEAT wait
550 LET y=CODE(INKEY$)
560 IF y=113 THEN EXIT wait
570 IF y=192 THEN EXIT wait
580 IF y=200 THEN EXIT wait
590 IF y=208 THEN EXIT wait
600 IF y=216 THEN EXIT wait
610 END REPEAT wait
620 END DEFine
```

```
630 DEFine PROCedure endcheck
640 FOR b=0 TO 3
650 FOR c=0 TO 2
660 IF a(b,c)=a(b,c+1) THEN LET w=1
670 IF a(c,b)=a(c+1,b) THEN LET w=1
680 NEXT c
690 NEXT b
700 IF w=0 THEN PRINT "YOU LOSE":STOP
710 END DEFine
720 DEFine PROCedure slip
730 FOR b=0 TO 3
740 FOR c=0 TO 2
750 FOR e=0 TO 2
760 IF a(b,c)=0 THEN LET s=1
770 IF s=1 THEN LET a(b,c)=a(b,c+1)
780 IF s=1 THEN LET a(b,c+1)=a(b,c+2)
790 IF s=1 THEN LET a(b,c+2)=a(b,c+3)
800 IF s=1 THEN LET a(b,c+3)=a(b,c+4)
810 IF s=1 AND a(b,c)>0 THEN LET n=1
820 IF s=1 AND a(b,c+1)>0 THEN LET n=1
830 IF s=1 AND a(b,c+2)>0 THEN LET n=1
840 LET s=0
850 NEXT e
860 NEXT c
870 NEXT b
880 END DEFine
```

```
890 DEFine PROCedure add
900 FOR b=0 TO 3
910 FOR c=0 TO 2
920 IF a(b,c)=a(b,c+1) THEN LET g=1
930 IF g=1 THEN LET a(b,c)=0
940 IF g=1 THEN LET a(b,c+1)=2*a(b,c+1)
950 IF g=1 THEN LET q=q+a(b,c+1)
960 IF g=1 THEN LET c=c+1
970 LET g=0
980 NEXT c
990 NEXT b
1000 END DEFine
1010 DEFine PROCedure transform
1020 FOR b=0 TO 3
1030 FOR c=0 TO 3
1040 LET d(b,c)=a(b,c)
1050 NEXT c
1060 NEXT b
1070 FOR b=0 TO 3
1080 FOR c=0 TO 3
1090 LET a(b,c)=d(c,3-b)
1100 NEXT c
1110 NEXT b
1120 END DEFine
1130 DEFine PROCedure inset
1140 FOR b=0 TO 3
```

```
1150 FOR c=0 TO 3
1160 IF a(b,c)=0 THEN LET z=z+1
1170 NEXT c
1180 NEXT b
1190 LET p=INT(RND(1,z))
1200 FOR b=0 TO 3
1210 FOR c=0 TO 3
1220 IF a(b,c)=0 THEN LET l=l+1
1230 IF l=p AND a(b,c)=0 THEN LET f=1
1240 IF f=1 THEN LET a(b,c)=2
1250 IF f=1 AND INT(RND(1,10))>8 THEN LET a(b,c)
=4
1260 LET f=0
1270 NEXT c
1280 NEXT b
1290 IF z=1 THEN endcheck
1300 LET l=0
1310 LET z=0
1320 LET w=0
1330 END DEFine
1340 DEFine PROCedure drawscreen
1350 CLS
1360 FOR b=0 TO 3
1370 FOR c=0 TO 3
1380 AT b,c*5:PRINT a(b,c)
1390 IF a(b,c)=0 THEN AT b,c*5:PRINT "-"
```

```
1400 NEXT c
1410 NEXT b
1420 AT 5,0:PRINT "score=";q
1430 END DEFine
```

COMMENT

GEORGE GWILT



I am reluctant to produce articles which contain coding since the resulting copy in QUANTA appears distorted. The two defects seem to result from word wrap when a line is too long for the chosen font and a peculiarity in the appearance of that font in the eMAG version.

I was surprised recently when, at a recent SQLUG meeting I compared a member's printed version with my eMAG version. The spacing of the letters, though acceptable in the printed version were distorted in the eMAG.

(Firstly apologies from me as Editor—This has caused a number of headaches in the past few issues of correctly placing characters where they should be in order to get listings to not only look right aesthetically but also correctly. The issues lie from a mix of a number of reasons, firstly font, font size, line spacing and number of characters per line. Secondly the algorithm used by the publisher program that squashes the spacing between characters

in order to fit within a space (it is not consistent with fixed proportion text used in listings).

Thirdly text codes used in QL systems and Windows systems behave differently CR, LF, soft returns and hard returns do not do as you would logically expect. Fourthly, the resolution between the printed magazine and the e-magazines are quite different and this affects the quality of both the text and the space between the text. Finally the readability factor, our readership need to be able see listings in order to understand them and copy them in to the QL system.

After discussing some of these with George, we have experimented with a number of these issues and we hope to have an improved listing especially for Assembler code in the next

RWAP Software

Replacement QL Keyboard Membranes

**Special Offer for Quanta Members
£12 each plus post and packing**

Quote "Quanta 2012 Offer"


Post and packing:
UK £2.50
Europe £2.80
Rest of World £3

Visit the QL Wiki for more details on software, books and hardware for the Sinclair QL:
www.rwapadventures.com/ql_wiki

RWAP Software
3 Dale View Court, Fulford, Stoke-on-Trent
Staffordshire ST11 9BA
Tel: 01782 398143
Email: rich@rwapservices.co.uk

UK Cheques payable to "R. Mellor" or use PayPal.

Website:
www.rwapsoftware.co.uk or www.sifmyretro.com



...skilled nanny
and some £1m...

ALICE
Allied services
on the £1m

Why a bad keyboard membrane could change your whole career strategy!

Working to safeguard the QL for the future.
www.rwapsoftware.co.uk

issue, watch this space—Ed.)

N.B.

If you have any comments on this point or any other issue about the magazine then please write/email to the editor , details on page 2 of this magazine - Ed.

NEMLUG NOTICES

SARAH GILPIN



meetings are held on the last Thursday of the month

(except August & December) from 7.00pm to 11.00pm. All are welcome to our meetings. The venue is usually at John Gilpin's home 181, Urmston Lane, Stretford, M/cr M32 9EH (near Junction 7 on M60 motorway) but currently at Alison Southern's home, 40, Distaff Road, Poynton, Cheshire, SK12 1HN. Contact us before the meeting – phone numbers inside front cover.

During 2016 there will be additional Sunday workshop meetings at 181, Urmston Lane on the following dates:

February 7th;
April 24th;
June 26th
Oct 30th

from 1.00pm to 7.00pm.

NEMQLUG - NORTH EAST MANCHESTER QL USER GROUP



n what was to be the last part of the SpriteDesigner we have nearly the rest of the SuperBASIC listing. The last part, should with any luck be in the next issue. It is littered with REM statements which hopefully explain and make sense of what is going on in each routine.

```
2120  REMark *****
2130  REMark The C key pressed invoking
2140  REMark the routine for changing colour
2150  REMark *****
2160  :
2170  IF KEYROW(2)=8
2180      ToColour fa
2190      Arrows
2200  END IF
2210  :
2220  REMark *****
2230  REMark The R key pressed invoking
2240  REMark the routine for Random Fill
2250  REMark *****
2260  :
2270  IF KEYROW(5)=16
```

```

2280     ToRandom
2290     Arrows
2300     END IF
2310 :
2320     IF t=0 AND Ta=0 : GO TO 1660
2330 :
2340     REMark *****
2350     REMark No key pressed from row 1
2360     REMark *****
2370 :
2380     IF KEYROW(1)>0 : GO TO 2380
2390 :
2400     REMark *****
2410     REMark Checks if SPACEBAR pressed
2420     REMark and toggles status of Pen
2430     REMark *****
2440 :
2450     IF t=64 THEN
2460         pen = NOT(pen) : Pend
2470     END IF
2480 :
2490     REMark *****
2500     REMark ENTER Key pressed and
2510     REMark sets a pixel of the
2520     REMark current selected colour
2530     REMark *****

```

```

2540 :
2550   IF t=1
2560     OVER #3,0 : STRIP #3,fa : INK #3,fa : AT
#3,y,x : PRINT #3,"w";
2570     INK #3,7 : OVER #3,-1 : p(x,y)=fa : BEEP
20,2
2580     BLOCK#1, Ste,1,(x+(8/Ste))
*Ste,y+16,fa
2590   END IF
2600   IF t=2 THEN
2610     IF x>0 : x=x-1
2620     IF x<0 : x=xm
2630   END IF
2640 :
2650   REMark *****
2660   REMark Cursor movement
2670   REMark *****
2680 :
2690   SElect ON t
2700     =4   : IF y>0 : y=y-1
2710     =8   : S : e : STOP           :REMark ESC
2720     =16  : IF x<xm+1 : x=x+1
2730     =128 : IF y<15 : y=y+1      :
2740   END SElect
2750 :
2760   IF pen=1 AND x<=xm THEN

```

```

2770     OVER #3,0 : STRIP #3,fa
2780     INK #3,fa : AT #3,y,x
2790     PRINT #3,"w"; : INK #3,7
2800     OVER #3,-1 : p(x,y)=fa
2810         BLOCK#1, Ste,1,(x+(8/Ste))
*Ste,y+16,fa
2820     BEEP 20,2
2830     END IF
2840 :
2850 REMark *****
2860 REMark *** TAB key pressed ****
2870 REMark *****
2880     IF Ta=8 THEN
2890         AT 0,0 : PRINT "TAB"
2900         x=xm+1
2910 :
2920 REMark *****
2930 REMark *** No key pressed ****
2940 REMark *****
2950     IF KEYROW(5)>0 : GO TO 2950
2960     END IF
2970 :
2980     IF x>xm : RETURN
2990     FOR w= 1 TO ti : END FOR w
3000 :
3010 REMark *****

```

```

3020 REMark *** No key pressed from row 0 ****
3030 REMark *****
3040 IF KEYROW(0)>0 : ti=ti-5 : ELSE : ti=50:
END IF
3050 GO TO 1660
3060 END DEFine
3070 :
3080 REMark
#####
3090 REMark # Selecting the sprite character
##
3100 REMark # and indicate whether selected
##
3110 REMark # or highlighted as the cursor moves
##
3120 REMark
#####
3130 :
3140 DEFine PROCedure SELECTing
3150 REMark #####
3160 REMark very basic key check
3170 REMark #####
3180 AT 0,0 : PRINT " ";
3190 IF KEYROW(1)>0 : GO TO 3180
3200 IF KEYROW(5)>0 : GO TO 3180
3210 OVER #4,-1
3220 sprx=0 : spry=0 : Ta=0

```

```

3230   bef=4 : DBef : bef=0
3240   PRINT #6,"SELECT SPRITE "; : Arrow
3250   PRINT #6,"TAB ^TAB"
3260   Message
3270   BLOCK #4,48,10,sprx*84,spry*20,7
3280   REPEAT stuff
3290       Ta=KEYROW(5): u=KEYROW(7)
3300       t=KEYROW(1)
3310       IF t=0 AND Ta=0 : GO TO 3290
3320       BLOCK #4,48,10,sprx*84,spry*20,7
3330       IF Ta=8 AND u=0           :
spx=2       :REMARK TAB
3340       IF t=4 AND spry>0       : spry=spry-1
3350       IF t=128 AND spry<9     : spry=spry+1
3360       IF t=2 : sprx=sprx-1
3370       IF t=16 : sprx=sprx+1
3380       IF sprx=2 OR sprx=-1    : EXIT
stuff :REMARK
3390       IF Ta=8 AND u=1:PleaseWait: EXIT
stuff :REMARK ^TAB
3400       IF t=1
3410           BLOCK #4,48,10,(sprn DIV 10)*84,(sprn
MOD 10)*20,4
3420           BLOCK #4,48,10,sprx*84,spry*20,4 :
sprn=sprx*10+spry
3430       END IF : REMARK *** Added during
reformat

```

```

3440      Message
3450      BLOCK #4,48,10,sprx*84,spry*20,7
3460      IF KEYROW(1)>0 : GO TO 3460
3470      END REPeat stuff
3480 END DEFine
3490 :
3500 REMark
#####
3510 REMark ## ToFill, fills the editing area
##
3520 REMark ## with the selected colour, you get
##
3530 REMark ## the option to cancel
##
3540 REMark
#####
3550 :
3560 DEFine PROCedure ToFill (colf)
3570 LOCAl x, y, t
3580      Mess "FILL WITH COLOUR "&colf
3590      BLOCK #6,508,10,0,0,3
3600      t=Cur_Pressed("12")
3610      IF t=1 : RETurn
3620      IF t=2
3630          PleaseWait
3640          OVER #3,0 : STRIP #3,colf : INK #3,colf
3650          FOR x=0 TO xm

```



```

3660      FOR y=0 TO 15
3670          AT #3,y,x : PRINT #3,"w"
3680          p(x,y)=colf
3690          BLOCK#1, Ste,1,(x+(8/Ste))
*Ste,y+16,colf
3700      END FOR y
3710  END FOR x
3720  INK #3,7 : OVER #3,-1
3730  Completed
3740  PAUSE 50
3750  END IF
3760  END DEFine
3770  :
3780  REMark *****
3790  REMark changes one colour to another
3800  REMark *****
3810  :
3820  DEFine PROCedure ToColour (fa)
3830  LOCAL t
3840  Mess "CHANGE COLOUR "&fa
3850  BLOCK #6,508,10,0,0,3
3860  t=Cur_Pressed("12")
3870  IF t = 1 : RETURN
3880  IF t = 2 :
3890      IF mo=4 THEN
3900          Messy "To Colour 0, 2, 4, 7"

```

```

3910         ELSE
3920             Messy "To Colour 0, 1, 2, 3, 4, 5, 6,
7"
3930         END IF
3940         Change_Colour fa, Key_Pressed
("01234567")
3950         Update_Matrix
3960     END IF
3970 END DEFine
3980 :
3990 REMark *****
4000 REMark Colour change routine
4010 REMark CFC - Change From Colour
4020 REMark CTC - Change To Colour
4030 REMark *****
4040 :
4050 DEFine PROCedure Change_Colour (CFC, CTC)
4060 LOCAL x, y
4070     FOR x = 0 TO xm
4080         FOR y = 0 TO 15
4090             IF p(x,y) = CFC THEN p(x,y) = CTC
4100         END FOR y
4110     END FOR x
4120 END DEFine
4130 :

```

The final part of the program will be in the next issue (hopefully)



**COPY DATE for APR/MAY 2016 ISSUE
5th APRIL 2016**

Please ensure that your copy reaches the editor on or before that date. Copy on Paper, Magnetic Media or by Email. Text & listings are preferred in ASCII format as text files, Microsoft™ Word Documents or in QL Quill formatted files. All graphic submissions in as high a resolution as possible, preferably in PNG format, other forms of submission can also be accepted, please contact the Editor for details.

All Contributions are welcome
Copy by Email to - editor@quanta.org.uk
Copy by Snail Mail, see page 2 for address

This magazine and all the articles etc. within it are copyright © QUANTA 2015 unless otherwise stated. Reproduction and redistribution in any form is not allowed without the express written permission of the publisher. All copyrights and trademarks are hereby acknowledged.

The online QL community

Now with QUANTA section!

New online chat facility!

Free to join!

No registration
needed to view

Please stop by for a visit!



www.qlforum.co.uk